

FIG. 1B

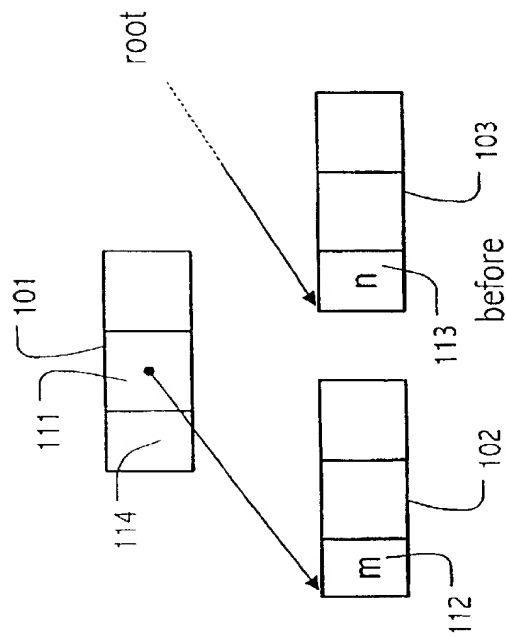


FIG. 1A

2/4

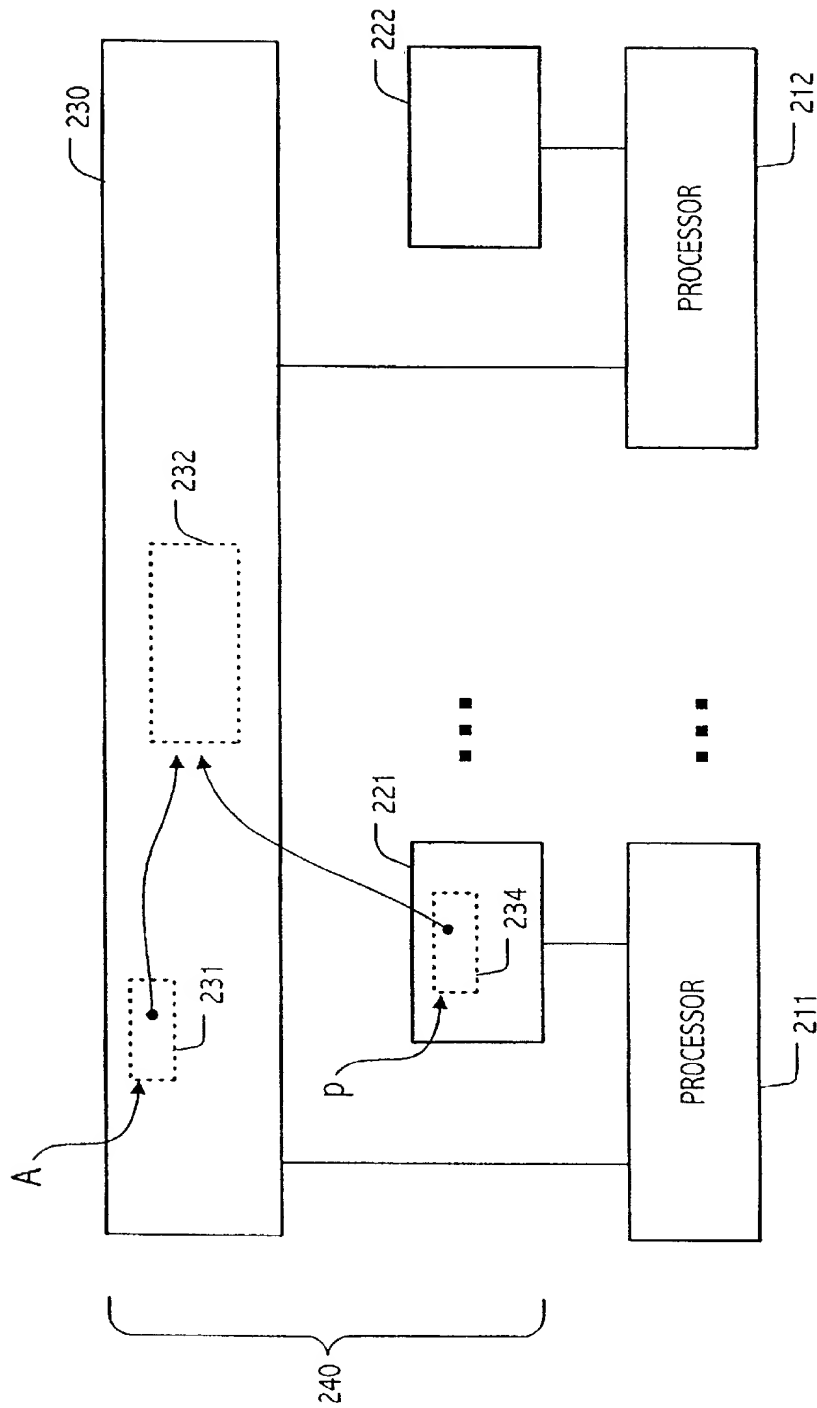


FIG. 2

3/4

```

class SNode {
1  class SNode *L, *R; valtype V; long rc;
2  SNode() {};
};

class Snark {
3  SNode *Dummy, *LeftHat, *RightHat;
4  Snark() {
        LeftHat(Null), RightHat(Null) {
35  LFRCHandleAlloc(&Dummy, new SNode);
36  LFRCHandle(&Dummy->L, Null);
37  LFRCHandle(&Dummy->R, Null);
38  LFRCHandle(&LeftHat, Dummy);
39  LFRCHandle(&RightHat, Dummy);
        };
40  ~Snark() {
41  while(popLeft() != EMPTY_val);
42  LFRCHandle(&Dummy, Null);
43  LFRCHandle(&LeftHat, Null);
44  LFRCHandle(&RightHat, Null);
        };
45  valtype pushRight(valtype v);
46  valtype pushLeft(valtype v);
47  valtype popRight();
48  valtype popLeft();
};

```

FIG. 4A

```

class SNode {
1  class SNode *L, *R; valtype V;
2  SNode() {};
};

class Snark {
3  SNode *Dummy, *LeftHat, *RightHat;
4  Snark() {
5      Dummy = new SNode;
6      Dummy->L = Dummy;
7      Dummy->R = Dummy;
8      LeftHat = Dummy;
9      RightHat = Dummy;
        };
10  valtype pushRight(valtype v);
11  valtype pushLeft(valtype v);
12  valtype popRight();
13  valtype popLeft();
};

```

FIG. 3A

4/4

```

valtype Snark::pushRight(valtype v) {
14  SNode *nd = new SNode;
15  SNode *rh = *rhR, *lh;
16  if (nd == Null)
17      return FULLval;
18  nd→R = Dummy;
19  nd→V = v;
20  while (true) {
21      rh = RightHat;
22      rhR = rh→R;
23      if (rhR == rh) {
24          nd→L = Dummy;
25          lh = LeftHat;
26          if (DCAS(&RightHat, &LeftHat, rh, lh, nd, nd))
27              return OKval;
28      } else {
29          nd→L = rh;
30          if (DCAS(&RightHat, &rh→R, rh, rhR, nd, nd))
31              return OKval;
32      }
33  }
34  }
35  }
36  }
37  }
38  }
39  }
40  }
41  }
42  }
43  }
44  }
45  }
46  }
47  }
48  }
49  SNode *nd = new SNode;
50  SNode *rh = Null, *rhR = Null, *lh = Null;
51  if (nd == Null) {
52      LFRCDestroy(rhR, nd, rh, lh);
53      return FULLval;
54  }
55  LFRCDStore(&nd→R, Dummy);
56  nd→V = v;
57  while (true) {
58      LFRCLoad(&RightHat, &rh);
59      LFRCLoad(&rh→R, &rhR);
60      if (rhR == Null) {
61          LFRCDStore(&nd→L, Dummy);
62          LFRCLoad(&LeftHat, &lh);
63          if (LFRCDCAS(&RightHat, &LeftHat,
64                      rh, lh, nd, nd)) {
65              LFRCDestroy(rhR, nd, rh, lh);
66              return OKval;
67          }
68      } else {
69          LFRCDStore(&nd→L, rh);
70          if (LFRCDCAS(&RightHat, &rh→R,
71                      rh, rhR, nd, nd)) {
72              LFRCDestroy(rhR, nd, rh, lh);
73              return OKval;
74          }
75      }
76  }
77  }
78  }
79  }
80  }
81  }
82  }
83  }
84  }
85  }
86  }
87  }
88  }
89  }
90  }
91  }
92  }
93  }
94  }
95  }
96  }
97  }
98  }
99  }
100 }
```

FIG. 3B

FIG. 4B